

---

---

# Introducción a Ruby on Rails

— Routes / Rake —

---

---

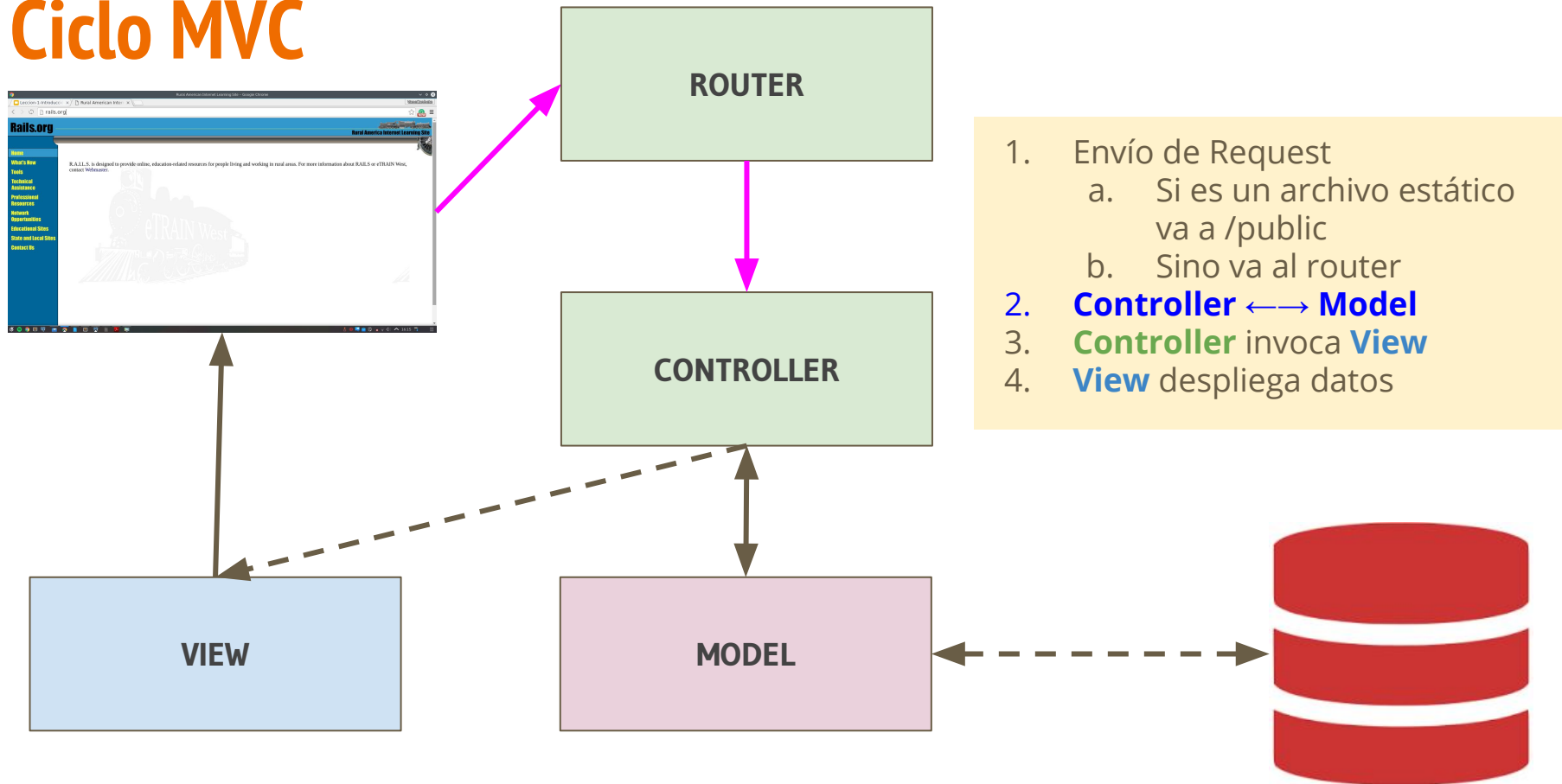
# Overview

- Cómo funciona el enrutamiento
- Rake
- Cómo analizar las rutas existentes en una aplicación

# Routes

- En la última lección **nos faltó un componente** para poder hacerlo funcionar. Este componente es el enrutamiento (routing).
- Antes de que un controller pueda **orquestrar a dónde debe ir una petición** web - esta petición **debe ser enrutada** al controller
- Pero, entonces cómo funcionó el código de **hello**?
  - Funcionó porque el route de la acción **hello** se generó automáticamente durante el **(rails g controller)**

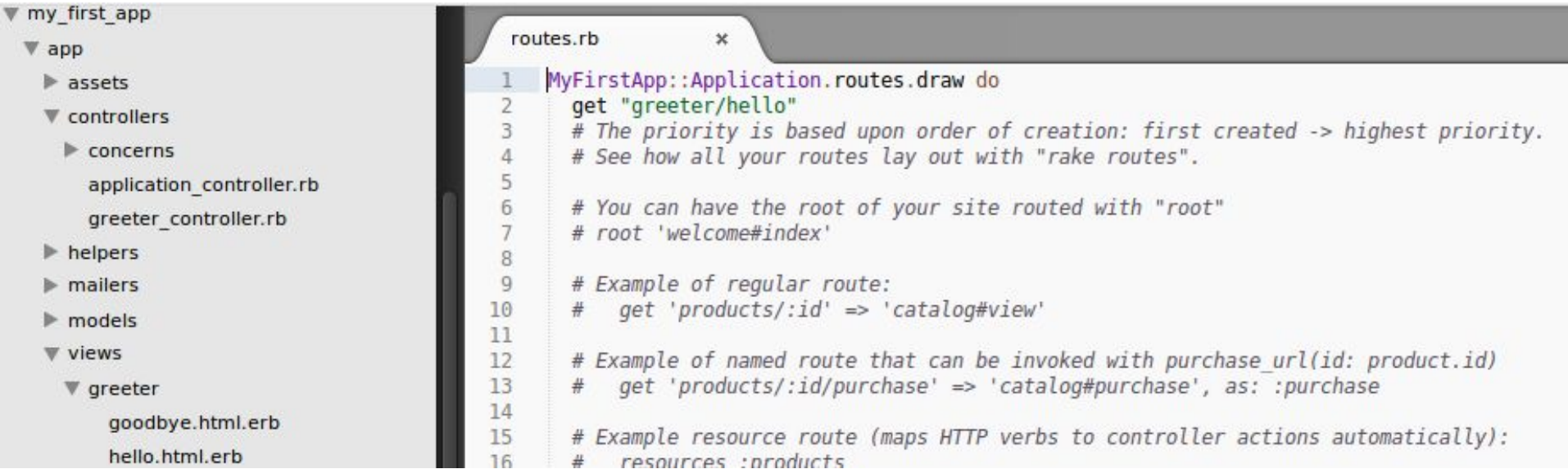
# Ciclo MVC



# ROUTES.RB

- Todos los routes **deben ser especificados** (sean generados por rails o sean agregados manualmente) en el archivo **config/routes.rb**.

# routes.rb



The screenshot shows a code editor interface. On the left is a file explorer for a project named 'my\_first\_app'. It contains a folder 'app' with subfolders 'assets', 'controllers', 'helpers', 'mailers', 'models', and 'views'. The 'controllers' folder is expanded, showing 'application\_controller.rb' and 'greeter\_controller.rb'. The 'views' folder is also expanded, showing 'goodbye.html.erb' and 'hello.html.erb'. On the right, the 'routes.rb' file is open. It contains the following code:

```
1 MyFirstApp::Application.routes.draw do
2   get "greeter/hello"
3   # The priority is based upon order of creation: first created -> highest priority.
4   # See how all your routes lay out with "rake routes".
5
6   # You can have the root of your site routed with "root"
7   # root 'welcome#index'
8
9   # Example of regular route:
10  # get 'products/:id' => 'catalog#view'
11
12  # Example of named route that can be invoked with purchase_url(id: product.id)
13  # get 'products/:id/purchase' => 'catalog#purchase', as: :purchase
14
15  # Example resource route (maps HTTP verbs to controller actions automatically):
16  # resources :products
```

greeter/hello → greeter\_controller / hello action

# Creemos el route para la acción goodbye

The image shows a code editor with two tabs: `routes.rb` and `goodbye.html.erb`. The `routes.rb` tab is active, displaying the following code:

```
1 MyFirstApp::Application.routes.draw do
2   #get "greeter/hello"
3
4   get 'greeter/hello' => "greeter#hello"
5   get 'greeter/goodbye'
6
```

Annotations above the code indicate the components of the route:

- A green box labeled **CONTROLLER** has an arrow pointing to the `greeter` part of the string `"greeter#hello"` in line 4.
- A green box labeled **ACTION** has an arrow pointing to the `hello` part of the string `"greeter#hello"` in line 4.

Below the code editor, a browser address bar shows the URL `localhost:3000/greeter/goodbye`.

**Goodbye, iván**

La hora es: 2016-10-30 21:40:52 -0300

# Rake

- Es el builder de Ruby.. (parecido al make de C, ant de Java)
- **Ruby's make**
- A diferencia de otros builders, no está escrito en XML o algún otro lenguaje de marcado, sino que está **hecho en Ruby**.
- Rails utiliza rake para **automatizar algunas tareas relacionadas a la aplicación**:
  - Bases de datos, ejecución de tests, etc.
- Para ver una **lista de las tareas de rake** (para las cuales tenemos soporte):
  - **rake --tasks**



# Rake tasks

```
→ my_first_app git:(master) X rake --tasks
rake about                                # List versions of all Rails frameworks and the environment
rake assets:clean[keep]                  # Remove old compiled assets
rake assets:clobber                       # Remove compiled assets
rake assets:environment                  # Load asset compile environment
rake assets:precompile                   # Compile all the assets named in config.assets.precompile
rake cache_digests:dependencies          # Lookup first-level dependencies for TEMPLATE (like messages/show or com
rake cache_digests:nested_dependencies  # Lookup nested dependencies for TEMPLATE (like messages/show or comments
rake db:create                           # Create the database from DATABASE_URL or config/database.yml for the cu
rake db:drop                             # Drops the database using DATABASE_URL or the current Rails.env (use db:
rake db:fixtures:load                    # Load fixtures into the current environment's database
rake db:migrate                          # Migrate the database (options: VERSION=x, VERBOSE=false, SCOPE=blog)
rake db:migrate:status                   # Display status of migrations
rake db:rollback                         # Rolls the schema back to the previous version (specify steps w/ STEP=n)
```

# Descripción de una tarea particular

- Si no tenemos ningún conocimiento de alguna tarea de rake y lo que hace, podemos pedir más información con el flag **--describe**
- **rake --describe routes**

```
→ my_first_app git:(master) x rake --describe routes  
rake routes
```

```
Print out all defined routes in match order, with names. Target specific controller with CONTROLLER=x.
```

# rake routes

- rake routes explica las rutas definidas actualmente en la aplicación.

```
→ my_first_app git:(master) x rake routes
```

	Prefix	Verb	URI Pattern	Controller#Action
greeter_hello		GET	/greeter/hello(.:format)	greeter#hello
greeter_goodbye		GET	/greeter/goodbye(.:format)	greeter#goodbye

## Entonces...

- El Router **direcciona** la petición web al **controlador apropiado**.
- **rake routes** permite ver las rutas definidas en la aplicación actual.